REMOVE

Vulnerable to TOCTOU issues

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-04

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 7247 bytes

Attack Category	• Path spoofing or confusion problem
Vulnerability Category	• Indeterminate File/Path • TOCTOU - Time of Check, Time of Use
Software Context	• File Management
Location	• stdio.h
Description	The remove() function makes a file or directory inaccessible by its name. An attempt to open the file or directory using that name does not work unless you recreate it. If the file is open, the subroutine does not remove it. If the file has multiple links, the link count of files linked to the removed file is reduced by 1. For files, remove() is identical to unlink(). For directories, remove() is identical to rmdir(). remove() is vulnerable to TOCTOU attacks. A call to remove() should be flagged if the first argument (the directory or file name) is used earlier in a check-category call.
APIs	**Function Name** / **Comments**: _remove — use; win32 _t — use; win32 _wremove — use; win32 remove — use unlink — rmdir —
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between

Function Name	Comments
_remove	use; win32
_t	use; win32
_wremove	use; win32
remove	use
unlink	
rmdir	

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

	the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results. The remove() call is a use-category call, which when preceded by a check-category call can be indicative of a TOCTOU vulnerability. A TOCTOU attack in regards to remove() can occur when a. A check for the existence of a file or other reference to a file or directory name occurs b. The file or directory is deleted Between a and b, an attacker could, for example, link the target file (the one to be removed) to a different known file. The subsequent removal of the target file would result in removal of the attacked file.
Exception Criteria	
Solutions	

Solution Applicability	**Solution Description**	**Solution Efficacy**
Generally applies to any remove().	Check that only the intended file was deleted.	Effective.
Generally applies to any remove().	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.
Generally applies to any remove().	Limit the interleaving of operations on files from	Does not eliminate the underlying vulnerability but can help

	multiple processes.	make it more difficult to exploit.
Generally applies to any remove().	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.

Signature Details	int remove(const char* FileName);

Examples of Incorrect Code	`#include <sys/types.h>`
	`#include <sys/stat.h>`
	
	`int check_status;`
	`int use_status;`
	`struct stat statbuf;`
	`...`
	
	`check_status=stat("toberemovedfile", &statbuf);`
	
	`...`
	`<long enough intervening code>`
	
	`use_status=remove("toberemovedfile");`

Examples of Corrected Code	
Source References	• Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X • man page for remove() • Microsoft Developer Network Library (MSDN)
Recommended Resource	
Discriminant Set	**Operating Systems**: • UNIX • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

1. mailto:copyright@cigital.com